

A Review On SLA And Various Approaches For Efficient Cloud Service Provider Selection

Shreyas G. Patel

Student of M.E, CSE Department, PIET Limda

Prof. Gordhan B. Jethava

Head & Assistant Professor, Information Technology Department, PIET Limda

Abstract

Cloud computing is becoming a popular choice for business to replace in-house IT Infrastructure. It delivers the IT enabled capabilities as a service to any needed users over the internet. It provides an on-demand, pay-as-you-go computing resources and economy of scale.

The increasing number of cloud providers, together with diverse type of services they offer on widely varying pricing schemes has lead to difficulties in comparing one cloud provider with another in terms of quality as well as cost of service which lead to complexities in cloud service selection. It is important to have a methodology such that the end user requirements might map to different criteria of the cloud services for selecting cloud services. This diversity in services and the number of available options have complicated the process of service and vendor selection for prospective cloud users and there is a need for a comprehensive methodology for cloud service selection. This paper focuses mainly on the different approaches that are existing, and comparative study of all the approach together as a literature survey.

1. Introduction

Cloud is a large pool of easily usable and accessible virtualized resources (Such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in

which guarantees are offered by the Infrastructure Provider by means of customized SLAs [1].

1.1 Classification of cloud

Cloud services have been classified into several categories on the basis of various technical and economic aspects. The first classification is based on service availability and second is based on application domain.

1.1.1 Based on availability of services

Public Cloud: In this category, cloud services are made available publicly to the users on pay-per-use basis. i.e. Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), SimpleDB, etc. [2].

Private Cloud: Here, cloud services are not available outside the peripheral of the organization. i.e. google uses Google File System and BigTable as cloud services internally [2].

Hybrid Cloud: When private cloud is not able to provide service to the coming request, it sends this request to some public cloud. These types of clouds are called Hybrid cloud [3].

1.1.2 Based on application domain

Infrastructure as a Service (IaaS) Cloud: Conventionally a cloud refers to an Infrastructure as a service cloud. Through virtualization, they are able to

split, assign and dynamically resize these resources to build ad-hoc systems as demanded by customers [2]. Platform as a Service (PaaS) Cloud: Cloud systems can provide the software platform where applications or services can run [2].

Software as a Service (SaaS) Cloud: This is an alternative to locally run applications. An example of this is the online alternatives of typical office applications such as word processors [2].

1.1.3 Features of cloud

Virtualization: It is a layer on the physical resources using which the optimum use of resources can be achieved. It provides virtual resources which may reside on a single physical machine or on several physical machines.

Dynamic Reconfiguration: Resources can be dynamically reconfigured to adjust to a variable load and provide optimum utilization of resources. Means it provides resources on demand as and when needed.

Scalability: Because of virtualization and the capability to add new virtual resources in resource pool, cloud provides scalability feature for large applications where load may increase at some time.

Pay-per-use: A cloud user just need to pay based on the time he uses the resources and amount of resources.

Service Level Agreement: SLA is an agreement between the cloud infrastructure service provider and consumer. It defines many things like the parties participating, lifetime of service, penalties, quality of service etc.

1.1.4 Issues in cloud computing

Lack of Standards: There are no specific standards in cloud to develop cloud-based applications and also defining Service Level Agreements [3]. These leads to some issues like [4] [5]:

- Migrating Application
- Migrating Data
- Moving application to cloud
- SLA matching and enforcement

Latency and Bandwidth: Because cloud services are often remote, they can suffer latency and bandwidth related issues associated with remote application [3].

Security: A user of a Cloud may store its data or deploy its software on Cloud, so a third party is

responsible to provide security of those software or data. Another security issue is that on a cloud two or more organizations may share the same physical resource and not be aware of it [3].

2. SLA in cloud computing

Quality attribute requirements play an important role in service-oriented architecture (SOA) environments. An SLA is part of the contract between the service consumer and service provider and formally defines the level of service.

Organizations seek to develop SLAs for various reasons. From a simple perspective, an SLA is developed between two parties to spell out who is responsible for what, what each party will do, and sometimes more importantly what each party will not do. Also an SLA defines the interaction between a cloud service provider and a cloud service consumer. An SLA contains several things [3]:

- A set of services the provider will deliver
- A complete, specific definition of each service
- The responsibilities of the provider and the consumer
- A set of metrics to determine whether the provider is delivering the service as promised
- An auditing mechanism to monitor the service
- The remedies available to the consumer and provider if the terms of the SLA are not met
- How the SLA will change over time

2.1 Machine-readable SLAs

Although managing and monitoring the quality levels of services rely heavily on automated tools, the actual SLA specification is typically a plain-text document, and sometimes it is an informative document published online. An example of a document with legal obligations is the Amazon S3 Service Level Agreement [6], Amazon EC2 Service Level Agreement [7].

SLAs are easy to create in plain text, but it is better to create them in a machine-readable format [3], such as XML, for these reasons [8]:

- A machine-readable format supports automatic matching of SLA template parameters between service users and providers.
- It also supports automatic negotiation between service users and providers.
- Sometimes the SLA specifies measures that should be taken by the service user and/or service provider when a deviation from the SLA

or a failure to meet the asserted service qualities occurs. In addition, machine-readable SLAs enable measures that can be triggered automatically (e.g., an email notification).

- A billing system can parse the SLA in order to obtain the rules to automatically calculate charges to the service user.
- An automated SLA management system that measures and monitors the quality parameters uses the SLA as input.

3. Related work

Cloud service selection is a highly important research issue but it has not received much research attention and little literature has been published in this area because cloud computing itself is still in its early stages. In this section we give a brief overview of the related work in cloud service selection.

Maximilien and Singh proposed a multi-agent based architecture to select the best service according to the consumers' preferences. Maximilien and Singh describe a system in which proxy agents gather information on services, and also interact with other proxy agents to maximize their information. [17] The model is described in the figure 3.1 given below. In the figure we can see that client interact with Cloud services with a proxy agent in between which collect and update its information about the Cloud services.

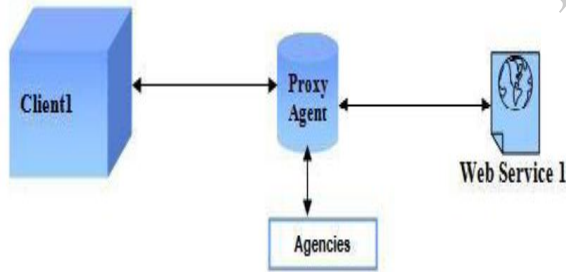


Fig 3.1. Model Proposed by Maximilien and Singh

Liu, Ngu, and Zeng proposed a model in which selection is done at the client side. Their major selection criteria is based on the QoS based service selection. They have considered three quality criteria namely execution time, execution duration and reputation for the selection. [18]. In addition, execution price, duration, transactions support, compensation and penalty rate are the other criteria. The authors of suggest an open, fair, and dynamic framework that evaluates the QoS of the available Cloud Services by using clients' feedback and monitoring. In figure 3.2 model proposed by Liu,

Ng and Zeng is shown. Here we can see that reasoning mechanism is attached at the client side means QoS parameters are applied at client side.

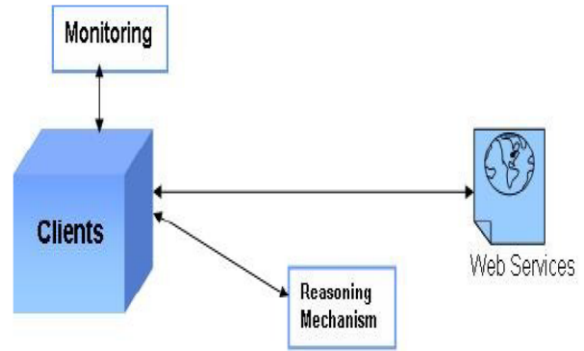


Fig 3.2. Model Proposed by Liu, Ngu and Zeng

Repository Model proposed by Abhishek Pandey and S.K. Jena uses a technique for dynamic selection of Cloud Services which will also handle the problem of redundant Cloud Services. This repository will be used to redirect the client's request. This will also provide a level of security since it will not be allowed to invoke directly by the clients. [19] This technique will prevent unauthorized access to the real services. This provision will also help to hide the systems complexity from the clients. The model is shown in the figure 3.3. In the figure we can see the repository between client and service and the QoS parameters are applied at the repository.

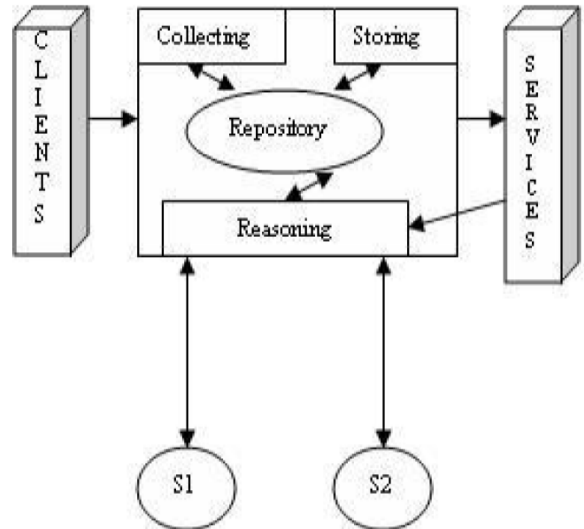


Fig 3.3. Repository Model

Significant level of research in SLAs has been performed during standardizing efforts. There are two

main specifications for describing an SLA for web services.

1) Web Service Agreement (WS-Agreement) [9] from Open Grid forum (OGF) and

2) Web Service Level Agreement language and framework (WSLA) [10] from IBM.

To the best of my knowledge, other research project for SLA specification includes:

SLAng: A language for SLAs which is part of the Trusted and Quality of Service Aware Provision of Application Services (TAPAS) project at University College London (UCL). [11]

Rule-Based Service Level Agreements (RBSLA) is a project that uses knowledge representation concepts for the specification of SLAs. [12]

Also, work is currently being done in the specification of service qualities that are part of SLAs.

Most advanced work in this area is the WS-Policy from the World Wide Web Consortium (W3C). By using WS-Policy, Service providers can advertise their policies. On the other hand service consumers can also specify their policy requirements [13].

Web Service Offering Language (WSOL), Web Service Modeling Ontology (WSMO) and Web Service Management layer (WSML) all provide some level of description for non-functional properties. However all the above work is not in the direct context of SLA [14].

4. Conclusion

This paper presents an overview of various approaches available for finding suitable service provider. Machine-readable, standardized SLAs are going to be an important element for organizations moving to third-generation, service-oriented systems characterized by the dynamic discovery, composition, and invocation of services based on QoS and other contextual information. There are ongoing efforts to codify and standardize SLAs for web services, in an effort to make SLAs machine readable. However, there is no established standard for SLA specification. Current tools and products for SLA management and specification mostly use their own proprietary formats. While this is indeed a step toward the automated negotiation and management of SLAs between service providers and consumers,

additional work in this area is needed to apply those tools in production environments.

5. Reference

- [1] G. Boss, P. Malladi, D. Quan, L. Legregni, and H. Hall, "Cloud Computing," IBM Corporation, 2007, pp. 1-17.
- [2] L.M. Vaquero, L. Rodero-merino, J. Caceres, and M. Lindner, "A Break in the Clouds : Towards a Cloud De_nition," ACM SIGCOMM Computer Communication Review, vol. 39, 2009, pp. 50-55.
- [3] Cloud Computing Use Case Discussion Group, "Cloud Computing Use Cases A white paper," vol. 4, 2010.
- [4] E.M. Maximilien, A. Ranabahu, R. Engehausen, and L. Anderson. "IBM altocumulus: a cross-cloud middleware and platform." In Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, pages 805-806. ACM, 2009.
- [5] Cirrocumulus: A Semantic Framework for Application and Core Services Portability Across Heterogeneous Clouds.
<http://knoesis.wright.edu/research/srl/projects/cirrocumulus/> (2010)
- [6] Amazon Simple Storage Service (S3) Service Level Agreement.
<http://aws.amazon.com/s3-sla/> (2007)
- [7] Amazon Elastic Compute Cloud (EC2) Service Level Agreement.
<http://aws.amazon.com/ec2-sla/> (2008)
- [8] P. Bianco, G.A. Lewis, and P. Merson, "Service Level Agreements in Service-Oriented Architecture Environments," TECHNICAL NOTE CMU/SEI-2008-TN-021, 2008.
- [9] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web Services Agreement Specification (WS-Agreement)," Global Grid Forum, 2005, pp. 1-80.
- [10] H. Ludwig, A. Keller, A. Dan, R.P. King, and R. Franck, "Web Service Level Agreement (WSLA) Language Specification," IBM Corporation, vol. 1, 2003, pp. 1-110.
- [11] Trusted and Quality of Service Aware Provision of Application Services (TAPAS). Overview.
<http://www.cs.ucl.ac.uk/staff/J.Skene/slang/> (2005).
- [12] Paschke, Adrian. RBSLA: Rule-Based Service Level Agreements.
<http://ibis.in.tum.de/projects/rbsla/index.php> (2006).
- [13] World Wide Web Consortium (W3C). Web Services Policy 1.5 – Framework.
<http://www.w3.org/TR/ws-policy/> (2007).

- [14] P. Patel, A. Ranabahu, and A. Sheth, "Service Level Agreement in Cloud Computing," OOPSLA 2009 - Workshop, 2009.
- [15] L. Green, "Service Level Agreements : An Ontological Approach," The Eighth International Conference on Electronic Commerce, 2006, pp. 185-194.
- [16] V. Stantchev and C. Schrpfer, "Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing," The 4th International Conference on Grid and Pervasive Computing, 2009, pp. 25-35.
- [17] E. Maximilien and M. Singh. " *Towards Autonomic Cloud Services, Trust and Selection.*" ICSOC'04 pages 212–221, November 2004.
- [18] Y. Liu, S. Ngu, and L. Zeng. " *QoS Computation and Policing in DynamicWeb Service Selection*" WWW2004, (8 pages), May 2004.
- [19] Abhishek Pandey, S.K.Jena " *Dynamic Approach for Cloud Services Selection*" Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong.

IJERT